

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Web

Membangun sebuah program berbasis web membutuhkan teori dan konsep yang kuat untuk mendukung kemudahan dalam merancang program sesuai dengan yang diharapkan. Konsep dasar web yang digunakan dalam penulisan Tugas Akhir ini adalah sebagai berikut:

2.1.1. Website

Menurut (Nurhadi, 2017, p. 8) mendefinisikan bahwa:

Website adalah suatu kumpulan-kumpulan halaman yang menampilkan berbagai macam informasi teks, data, gambar diam maupun bergerak, data animasi, suara, video maupun gabungan dari semuanya yang membentuk satu rangkaian bangunan yang saling berkaitan dimana masing-masing dihubungkan dengan jaringan halaman (*hyperlink*).

Website dapat bersifat statis maupun dinamis yang biasanya terangkum dalam sebuah domain maupun subdomain, tepatnya berada dalam WWW (*World Wide Web*) yang tentunya berada dalam internet.

Dalam membangun program atau aplikasi berbasis web ada beberapa tools yang dibutuhkan, diantaranya:

1. Internet

Menurut (Wijaya, 2015, p. 1) menyatakan bahwa, “internet (*Interconnected Network*) adalah sebutan untuk jaringan komputer global

yang menghubungkan satu komputer dengan komputer lain yang ada di seluruh dunia”.

2. Web Browser

Web browser digunakan untuk menampilkan hasil website yang telah dibuat. Dalam pembuatan Tugas Akhir ini menggunakan lebih dari satu web browser, tujuannya untuk memastikan desain website dapat tampil sempurna di berbagai web browser.

3. Web Server

Dalam membangun sebuah website dibutuhkan Apache Server dan Phpmyadmin. Menurut bahwa, “Apache adalah aplikasi yang berfungsi sebagai web server, fungsinya sebagai tempat menyimpan file-file PHP (*Hypertext Preprocessor*) dan file-file lainnya yang diperlukan website”. Sedangkan Phpmyadmin menurut (Abdulloh, 2016, p. 5) “adalah aplikasi berbasis web yang digunakan untuk membuat basis data (*Database*) MySQL sebagai tempat untuk menyimpan data-data website”. Keduanya aplikasi tersebut sudah ada dalam paket aplikasi XAMPP.

4. XAMPP

XAMPP adalah perangkat lunak bebas yang mendukung banyak sistem operasi yang merupakan komposisi dari berbagai program diantaranya Apache, MySQL, PHP, dan Perl. Saat ini XAMPP sudah memasuki versi 3.2.2 dengan PHP versi 7.3.6 dan merupakan versi yang dipakai dalam pembuatan program untuk Tugas Akhir ini.

5. Text Editor

Text Editor digunakan untuk menulis skrip HTML (*Hyper Text Markup Language*), CSS (*Cascading Style Sheets*), PHP (*Hypertext Preprocessor*), Javascript dan berbagai bahasa pemrograman lainnya. Berbagai text editor banyak digunakan namun dalam penulisan program untuk Tugas Akhir ini penulis menggunakan Sublime Text karena lebih mudah dan familier bagi penulis dibanding text editor lainnya. Sublime Text sendiri sudah memasuki versi 3.2.1 Build 3270 yang sudah mengalami perbaikan sehingga dapat memaksimalkan dalam pembuatan kode program.

2.1.2. Bahasa Pemrograman

Menurut (Abdulloh, 2016, p. 1) “Bahasa pemrograman merupakan bahasa yang dapat dipahami oleh komputer”. Dalam membangun sebuah program web definisi jenis bahasa yang digunakan dalam pembuatan Tugas Akhir ini adalah sebagai berikut:

1. *Hyper Text Markup Language (HTML)*

Menurut (Abdulloh, 2016, p. 2), “HTML atau singkatan dari *Hyper Text Markup Language*, yaitu skrip yang berupa tag-tag untuk membuat dan mengatur struktur website”. Beberapa tugas utama HTML dalam membangun website menurut (Abdulloh, 2016, p. 2) diantaranya sebagai berikut:

- a. Menentukan layout website.
- b. Memformat text dasar, seperti pengaturan paragraph dan format font.

- c. Membuat list.
- d. Membuat tabel.
- e. Menyisipkan gambar, video dan audio.
- f. Membuat link.
- g. Membuat formulir.

2. *Cascading Style Sheets (CSS)*

Menurut (Abdulloh, 2016, p. 2), “CSS singkatan dari *Cascading Style Shessts*, yaitu skrip yang digunakan untuk mengatur desain website”. Fungsi CSS memberikan pengaturan yang lebih lengkap dibanding HTML, struktur website yang dibuat dengan HTML sebelumnya akan terlihat rapih dan elegan dengan penambahan skrip CSS. CSS memiliki *framework* yang bernama Bootstrap yang berisi sekumpulan file CSS, font dan javascript yang siap diintegrasikan ke sebuah dokumen HTML menggunakan kaidah tertentu. (Subagia, 2018, p. 53).

3. *Hypertext Preprocessor (PHP)*

Bahasa pemograman yang dibuat oleh Rasmus Lerdorf pada tahun 1995 ini awalnya bernama *Personal Home Page*, lalu pada tahun 1997 perusahaan bernama Zend menulis ulang interpreter PHP menjadi bersih, lebih baik dan lebih cepat. Pada tahun 1998 dirilis PHP 3.0 dan merubah akronim PHP menjadi *Hypertext Preprocessor*. Menurut (Subagia, 2018, p. 1) menjelaskan bahwa, “PHP (*Hypertext Preprocessor*) adalah bahasa pemograman yang berjalan dalam sebuah web server (*server side*)”. Fungsinya adalah untuk melakukan pengolahan data pada database. Data

website akan dimasukan ke database, diedit dihapus dan ditampilkan pada website yang diatur oleh PHP.

4. *Framework CodeIgniter (CI)*

Menurut (Supono & Putratama, 2016, p. 109) menyatakan bahwa, “*Framework* secara sederhana dapat diartikan kumpulan dari fungsi-fungsi atau prosedur-prosedur dan class-class untuk tujuan tertentu yang sudah siap untuk digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang pemogram, tanpa harus membuat fungsi atau *class* dari awal”. Menurut (Supono & Putratama, 2016, p. 109) bahwa, “Codeigniter adalah aplikasi open source berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun website dinamis dengan menggunakan PHP”. *Codeigniter* memudahkan *developer* atau pengembang web untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuat dari awal.

5. Javascript

Menurut (Subagia, 2018, p. 14) menyatakan bahwa, “Javascript merupakan bahasa pemograman web yang bersifat *Client Side Programming*, artinya pemosesan dilakukan disisi klien”. Berbeda dengan PHP yang di proses disisi *server*, javascript diproses di sisi *client*, sehingga membuat javascript lebih interaktif dibanding PHP. Peran javascript adalah memberikan efek animasi yang menarik dan interaktivitas dalam penanganan *event* yang dilakukan oleh pengguna website.

6. *JQuery*

Menurut (Abdulloh, 2016, p. 147) mendefinisikan bahwa, “*JQuery* merupakan salah satu javascript *library*, yaitu kumpulan fungsi javascript siap pakai, sehingga mempermudah dan mempercepat kita dalam membuat kode javascript”. Penggunaan *JQuery* tidak perlu mengetikkan kode javascript secara panjang, hanya tinggal memanggil fungsi yang sudah disediakan untuk menghasilkan output yang sama. *JQuery* juga dapat digunakan untuk mengatur HTML, CSS, dan animasi sehingga desain lebih menarik.

2.1.3. Basis Data (*Database*)

Menurut (Indrajani, 2015, p. 70) menyatakan dalam bukunya bahwa, “Basis data adalah sebuah kumpulan data yang saling berhubungan secara logis, dan merupakan sebuah penjelasan dari data tersebut, yang didesain untuk menemukan data yang ditentukan oleh sebuah organisasi”.

Menurut (Priyadi, 2014, p. 2) menyatakan bahwa, “basis data adalah sekumpulan fakta berupa representasi table yang saling berhubungan dan disimpan dalam media penyimpanan secara digital”.

Basis data terdiri dari tabel saling berelasi dan tidak berelasi dan semua data diintegrasikan dengan menghindari duplikasi data. Basis data dapat digunakan oleh banyak departemen dan pemakai. Basis data tidak hanya memegang data operasional organisasi, tetapi juga penjelasan mengenai data tersebut. Karena alasan tersebut basis data dapat juga dideskripsikan sebagai

kumpulan data yang saling terintegrasi. Dalam basis data terdapat 3 elemen diantaranya:

1. *Tabel*

Pada suatu basis data, *tabel* direpresentasikan menjadi suatu bentuk segiempat berupa matriks, yang terdiri dari kolom dan baris.

2. *Field*

Field atau biasa disebut kolom yaitu merupakan representasi dari sebuah nama kolom tersebut yang pembacaan datanya dilakukan secara vertikal

3. *Record*

Record adalah data dalam satu baris atau dibaca secara horizontal. Nilai-nilai dari sebuah *record* merupakan gabungan dari semua *field* yang terdapat dalam tabel tersebut.

Perangkat lunak yang digunakan untuk mengelola basis data disebut dengan DBMS (*Database Management System*). Menurut (Trisyanto, 2017, p. 122) bahwa, “DBMS adalah *software* yang menangani semua akses ke basis data”.

Salah satu perangkat lunak pengelola basis data adalah *MySQL* yang penulis juga pakai sebagai DBMS (*Database Management System*) pada pembuatan program untuk Tugas Akhir ini.

MySQL dikembangkan oleh sebuah perusahaan di Swedia yang bernama *MySQL AB* dengan istilah T.c.X yang dibuat sekitar tahun 1994-1995.

MySQL merupakan perkembangan dari bahasa SQL (*Structured Query Language*) yaitu bahasa terstruktur yang digunakan untuk membangun basis data dan interaksi antar *script* program dengan *database* server dalam hal pengolahan data.

Proses pembangunan aplikasi basis data, perintah SQL dibagi menjadi 3 jenis, yaitu:

1. *Data Definition Language* (DDL)

Data Definition Language (DDL) merupakan perintah SQL yang digunakan untuk melakukan definisi awal suatu basis data dan table pada konsep RDMS (*Relational Database Management System*). Contoh perintah yang digunakan yaitu: CREATE, DELETE dan ALTER.

2. *Data Manipulation Language* (DML)

Data Manipulation Language (DML), merupakan perintah SQL yang digunakan untuk melakukan pengolahan *record* atau manipulasi *record* pada table dalam suatu basis data. Contoh perintah yang digunakan yaitu: INSERT, UPDATE, SELECT, DELETE.

3. *Data Control Language* (DCL)

Data Control Language (DCL) merupakan perintah SQL yang digunakan untuk melakukan pengaturan hak akses suatu objek data para pengguna dalam basis data. Contoh perintah yang digunakan adalah: GRANT dan REVOKE.

2.1.4. Model Pengembangan Perangkat Lunak

“Tahapan pengembangan perangkat lunak atau SDLC (*System Development Life Cycle*) adalah proses pengembangan atau mengubah suatu sistem perangkat lunak dengan menggunakan model yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya” menurut (Sukamto & Shalahuddin, 2018, p. 26). Dalam pembuatan program ini penulis menggunakan model *Water Fall*. “Model air terjun (*water fall*) menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahapan pendukung (*support*)” menurut (Sukamto & Shalahuddin, 2018, p. 28). Tahapan pengembangan model *water fall* menurut (Sukamto & Shalahuddin, 2018, p. 28) adalah sebagai berikut:

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

3. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung (*support*) atau Pemeliharaan (*maintenance*)

Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.2. Teori Pendukung

Teori pendukung dalam merancang program untuk Tugas Akhir ini adalah sebagai berikut:

2.2.1. Rancangan Dokumen

Menurut (Lestanti & Susana, 2016) menjelaskan bahwa, “Dokumen adalah surat penting atau berharga yang sifatnya tertulis atau tercetak yang berfungsi atau dapat dipakai sebagai bukti ataupun keterangan”.

Dalam sebuah sistem berjalan membutuhkan sebuah dokumen-dokumen yang dijadikan sebagai informasi dalam mengelola sistem tersebut. Dokumen bisa berupa dokumen masuk dan dokumen keluar yang keduanya dijelaskan secara rinci dengan uraian parameter-parameter tertentu.

2.2.2. Struktur Navigasi

Pada Pengembangan aplikasi berbasis *web*, tentunya dalam membuat lebih dari satu halaman web atau bahkan bisa ratusan jika aplikasi yang dibangun kompleks membutuhkan fitur navigasi.

Menurut (Ardiansyah, 2016, p. 61) menyatakan bahwa, “Struktur navigasi dapat diartikan sebagai alur suatu program yang menggambarkan rancangan hubungan antara area yang berbeda sehingga memudahkan proses pengorganisasian seluruh elemen-elemen *website*”. Struktur navigasi terbagi menjadi lima diantaranya sebagai berikut:

1. Struktur Navigasi linier

Struktur navigasi linier hanya mempunyai satu rangkaian cerita yang berurut, yang menampilkan satu demi satu tampilan layar secara berurut menurut urutannya. Tampilan yang dapat ditampilkan pada struktur jenis ini adalah satu halaman sebelumnya atau satu halaman sesudahnya, tidak dapat dua halaman sebelumnya atau dua halaman sesudahnya.

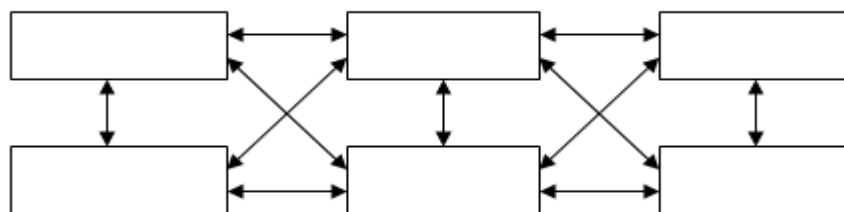


Sumber: (Ardiansyah, 2016)

Gambar II.1. Struktur Navigasi Linier

2. Struktur Navigasi Non-linier

Struktur navigasi non-linier atau struktur tidak berurut merupakan pengembangan dari struktur navigasi linier. Pada struktur ini diperkenankan membuat navigasi bercabang. Percabangan yang dibuat pada struktur nonlinier ini berbeda dengan percabangan pada struktur hirarki, karena pada percabangan nonlinier ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama yaitu tidak ada *Master Page* dan *Slave Page*.

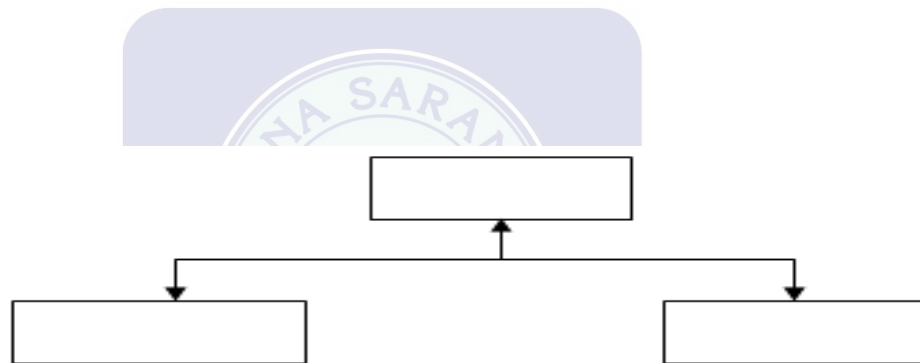


Sumber: (Ardiansyah, 2016)

Gambar II.2. Struktur Navigasi Non-Linier

3. Struktur Navigasi Hirarki

Struktur navigasi hirarki atau struktur bercabang, yaitu suatu struktur yang mengandalkan percabangan untuk menampilkan data berdasarkan kriteria tertentu. Tampilan pada menu pertama disebut sebagai *Master Page* (halaman utama pertama), halaman ini mempunyai halaman percabangan yang disebut *Slave Page* (halaman pendukung). Jika salah satu halaman pendukung dipilih atau diaktifkan, maka tampilan tersebut akan bernama Master Page (halaman utama kedua), dan seterusnya.



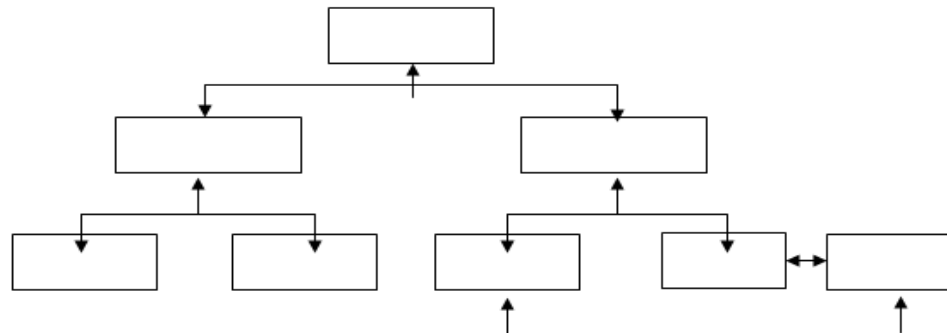
Sumber: (Ardiansyah, 2016)

Gambar II.3. Struktur Navigasi Hirarki

4. Struktur Navigasi *Composite* (Campuran)

Struktur navigasi *Composite* campuran merupakan gabungan dari ketiga struktur sebelumnya yaitu linier, non-linier dan hirarki. Struktur navigasi ini juga biasa disebut dengan struktur navigasi bebas. Struktur navigasi ini banyak digunakan dalam pembuatan website karena struktur ini dapat

digunakan dalam pembuatan website sehingga dapat memberikan ke-interaksian yang lebih tinggi.



Sumber: (Ardiansyah, 2016)

Gambar II.4. Struktur Navigasi Campuran

2.2.3. *Entity Relationship Diagram (ERD)*

Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan ERD (*Entity Relationship Diagram*). Menurut (Indrajani, 2015, p. 33) bahwa “ERD (*Entity Relationship Diagram*) adalah sebuah pendekatan *top-bottom* dalam perancangan basis data yang dimulai dengan mengidentifikasi data-data terpenting yang disebut dengan entitas dan hubungan antara entitas-entitas tersebut digambarkan dalam suatu model”.

Menurut (Trisyanto, 2017, p. 89) menyatakan bahwa “ERD (*Entity Relationship Diagram*) adalah gambar atau diagram yang dibuat, disimpan dan digunakan dalam sistem bisnis”.

ERD (*Entity Relationship Diagram*) digunakan untuk pemodelan basis data relasional, sehingga jika penyimpanan basis data menggunakan OODBMS (*Object Oriented Database Management System*) maka perancangan basis data

tidak perlu menggunakan ERD. ERD (*Entity Relationship Diagram*) memiliki aliran notasi seperti notasi *Chen* yaitu notasi yang dikembangkan oleh Peter Chen. Berikut adalah simbol-simbol yang digunakan ERD dengan notasi *Chen* dalam (Sukamto & Shalahuddin, 2018, p. 50):

1. Entitas (*Entity*)

Entitas merupakan data inti yang akan disimpan atau objek yang mewakili suatu yang nyata dan dapat dibedakan dengan suatu yang lain. Penamaan entitas biasanya lebih kepada kata benda dan belum merupakan nama table. Simbol dari entitas biasanya digambarkan dengan persegi panjang.

2. Atribut

Setiap entitas memiliki elemen yang disebut atribut yaitu *field* atau kolom data yang butuh disimpan dalam suatu entitas. Simbol atribut digambarkan dengan oval.

3. Atribut Kunci Primer

Atribut kunci primer digunakan untuk menandakan kunci akses untuk *record* yang diinginkan. Atribut kunci primer ini disimbolkan dengan oval yang digaris bawah pada nama *field*.

4. Atribut Multinilai

Atribut multinilai yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu. Simbolnya dilambangkan dengan oval di dalam oval.

5. Relasi

Penghubung antar entitas disebut dengan relasi. Biasanya diawali dengan kata kerja. Simbolnya dilambangkan dengan belah ketupat. Terdapat tiga buah jenis relasi pada table di dalam bagan ERD, yaitu:

a. *One to One*

Relasi ini menggambarkan hubungan satu himpunan entitas pada table pertama ke satu himpunan entitas pada table kedua.

b. *One to Many*

Relasi ini menggambarkan hubungan satu himpunan entitas pada table pertama ke banyak himpunan entitas pada table kedua.

c. *Many to Many*

Relasi ini menggambarkan hubungan banyak himpunan entitas pada table pertama ke banyak himpunan entitas pada table kedua.

6. Asosiasi (*Assosiation*)

Asosiasi digunakan untuk penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian. Simbol dari asosiasi ini dilambangkan dengan garis lurus.

2.2.4. *Logical Record Structure (LRS)*

Menurut Wijaya dan Sari dalam (Imaniawan & Nur, 2019) menyatakan bahwa, “*Logical Record Structure (LRS)* adalah representasi dari struktur *record-record* pada table-tabel yang terbentuk dari hasil antar himpunan entitas”.

Dari penjelasan diatas dapat disimpulkan bahwa *Logical Record Structure (LRS)* adalah struktur dari tipe record pada table-tabel antar himpunan entitas yang merupakan penjabaran lebih lanjut dari ERD (*Entity Relationship Diagram*).

2.2.5. Spesifikasi File

Menurut (Subandi & Syahidi, 2018, p. 24) mengatakan bahwa, “Aktivitas perancangan basis data akan mentranformasikan spesifikasi kebutuhan untuk tempat penyimpanan data yang dikembangkan selama analisis basis data ke dalam spesifikasi terstruktur untuk memandu implementasi langsung ke dalam sistem basis data yang digunakan”.

Spesifikasi file ini menggambarkan struktur data fisik pada suatu sistem atau aplikasi. Spesifikasi ini menyajikan bagaimana penyimpanan data dilakukan di perangkat lunak basis data.

2.2.6. Pengkodean

Menurut (Mustakini, 2014, p. 384) pengkodean adalah “Suatu susunan digit (angka), huruf dan karakter-karakter khusus yang dapat dirancang dalam bentuk kode”.

Kode digunakan untuk mengklarifikasi data, memasukan data ke dalam ke dalam komputer dan untuk mengambil bermacam-macam informasi yang berhubungan dengannya. Kode dapat berupa angka, huruf dan karakter khusus. Manfaat pengkodean antara lain:

1. Mempercepat atau mempersingkat proses penulisan baik dari elemen data, proses penyajian maupun peng-entry-an data pada komputer.

2. Menghemat media penyimpanan data seperti harddisk dan memori.
3. Untuk mempermudah dan mempercepat proses pemasukan, pencarian serta pengolahan data guna memperoleh informasi yang akurat.

2.2.7. Implementasi dan Pengujian Web

Setelah membuat sebuah sistem informasi langkah selanjutnya adalah mengimplementasikan sistem tersebut menjadi sebuah perangkat lunak, kemudian dilakukannya sebuah tahap pengujian terhadap perangkat lunak tersebut agar dapat diketahui sejauh mana sistem informasi yang telah diimplementasikan tersebut memiliki nilai (manfaat) bagi *user* dalam menyelesaikan masalah yang ada.

Menurut (Sukanto & Shalahuddin, 2018, p. 272) menyatakan bahwa “Pengujian perangkat lunak adalah sebuah elemen sebuah topik yang memiliki cakupan luas dan sering dikaitkan dengan verifikasi (*verification*) dan validasi (*validation*) (V&V)”.

Terdapat empat tahap pengujian yang berkaitan dengan verifikasi menurut (Sukanto & Shalahuddin, 2018, p. 274) diantaranya:

1. Pengujian Unit

Pengujian unit memiliki keterkaitan pemograman terstruktur contohnya saat program menuliskan atau membaca data di basis data.

2. Pengujian Integrasi

Pengujian ini merupakan gabungan dari dua atau lebih unit pada perangkat lunak.

3. Pengujian Sistem

Pengujian sistem adalah dimana unit-unit proses yang sudah diintegrasikan diuji dengan antarmuka yang sudah dibuat sehingga pengujian ini dimaksudkan untuk menguji sistem perangkat lunak secara keseluruhan dan diuji secara satu sistem (tidak terpisah-pisah lagi).

4. Pengujian Penerimaan

Setelah pengujian sistem selesai langkah berikutnya adalah pengujian penerimaan dimana sistem akan diuji oleh pelanggan (*customer*) atau user (pemakai perangkat lunak). Pengujian penerimaan digunakan untuk mengetahui kepuasan pelanggan atau *user* terhadap perangkat lunak yang telah dibuat.

Pengujian yang berkaitan dengan validasi memiliki dua pendekatan menurut (Sukanto & Shalahuddin, 2018, p. 275) yaitu:

1. Pengujian Kotak Hitam (*Black-Box Testing*)

Black-Box Testing adalah pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

2. Pengujian Kotak Putih (*White-Box Testing*)

White-Box Testing adalah pengujian dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi masukan dan keluaran yang sesuai dengan spesifikasi kebutuhan. Pengujian kotak putih dilakukan

dengan memeriksa logik dari kode program misalkan menguji alur pengulangan (*looping*) dan seleksi kondisi pada logika pemograman yang dibuat.

2.3. Complaint Online

2.3.1. Komplain (*Complaint*)

Menurut (Stauss & Seidel, 2018, p. 26) menyatakan bahwa “Komplain adalah ekspresi ketidakpuasan”. Ketidakpuasan tersebut diekspresikan melalui lisan atau tulisan yang menyatakan pengadu atau pembuat komplain menyatakan tidak puas atau mengalai suatu kendala. (Stauss & Seidel, 2018, p. 26).

Menurut (Stauss & Seidel, 2018, p. 22) menyatakan juga bahwa, “Komplain merupakan penanda indikator masalah dan sebagai masukan pelanggan”.

Komplain juga bisa dikatakan sebagai interaksi antara pemakai jasa atau produk dan penyedianya yang merupakan bagian dari bentuk komunikasi”. Komplain atau keluhan sangat dibutuhkan karena akan menghasilkan informasi baik positif maupun negatif tentang ketidaksesuaian yang dirasakan pihak kedua yang menerima jasa atau produk. Bahkan komplain itu merupakan sebuah komunikasi aktif yang bisa menjurus ke dalam sebuah “interaksi” (Abdussalaam & Anggieta Saputra, 2018).

Kebutuhan akan manajemen komplain diperlukan untuk dapat menangani keluhan dengan lebih baik dan ini sekaligus menunjukkan pentingnya perhatian dan

kepedulian penyedia jasa atau produk kepada pelangganya. Menurut (Mursalim, 2018) menyatakan bahwa, “Manajemen pengaduan (komplain) yaitu suatu sistem yang digunakan untuk memonitor sikap dan kepuasan para pelanggan, penyalur, dan partisipan lain dalam sistem pemasaran sehingga ketika terjadi permasalahan manajemen dapat mengambil langkah yang lebih cepat untuk menyelesaikannya”.

2.3.2. Daring (Online)

Menurut (Kemenristekdikti, 2017) mendefinisikan bahwa, “Daring, atau dalam jaringan adalah terjemahan dari istilah online yang bermakna tersambung ke dalam jaringan komputer”.

Secara umum daring atau *online* menunjukkan keadaan terhubung. Dalam KBBI (*Kamus Besar Bahasa Indonesia*) daring merupakan kependekan dari kata dalam jaringan. Secara tidak langsung *online* juga dapat diartikan sebagai suatu keadaan komputer yang dapat saling bertukar informasi karena sudah terhubung.

2.3.3. Complaint Online

Berdasarkan penjelasan komplain (*complaint*) dan daring (*online*) bisa dikatakan bahwa *Complaint Online* adalah keluhan yang disampaikan lewat sebuah jaringan yang saling terhubung melalui media gadget berupa komputer, ponsel pintar (*smartphone*) dan lain-lain.